

1.БАЗИС. СТРУКТУРА И ОСНОВНЫЕ ПРИНЦИПЫ

Графтор представляет собой библиотеку подпрограмм и функций на фортране и служит интерфейсом между прикладной программой и конкретным графическим устройством. Такой интерфейс позволяет перейти от краткого задания сложного комплексного изображения к немногочисленным и весьма простым командам устройства.

В библиотеке Графтора более 400 программных компонент, каждая из которых относится либо к базисному уровню, либо к одной из функциональных групп. Базисный уровень определяет общую организацию библиотеки и ее мобильность по отношению к разнообразным графическим устройствам. В базисный уровень входит набор графических примитивов, к которым относятся отрезки прямых линий, дуги окружностей и эллипсов, многоугольники, числовые и текстовые цепочки, маркеры. Базисный уровень служит основой при разработке функциональных групп, каждая из которых несет очевидную предметную окраску.

1.1. Основные понятия

Решение любой задачи с помощью ЭВМ - это, по существу, моделирование некоторого явления или процесса. Прикладная программа, которая осуществляет такое моделирование, оперирует с объектами, представленными в памяти машины некоторыми структурами данных.

К наиболее простым и распространенным объектам можно отнести табулированную функцию одного переменного, представленную двумя одномерными массивами (значениями абсцисс и ординат). Функция двух переменных (поверхность), как правило, представляется матрицей и, возможно, двумя одномерными массивами, задающими сетку. В некоторых случаях, когда сетка равномерная (например, шаг по оси X постоянен), бывает достаточно и одного массива.

В системах машинного проектирования объекты имеют иную природу, и может потребоваться более сложное представление с привлечением таких структур, как списки и записи. В самом деле, элементы электронной схемы (скажем, диоды, резисторы и т.п.) имеют несколько разнородных характеристик, а топология самой схемы может быть довольно сложной и определяться связями между элементами.

Далее нас будут интересовать те из объектов, которые требуется получить в виде графического образа на *видовой поверхности*, т.е. на бумаге, экране дисплея, кинолентке или на

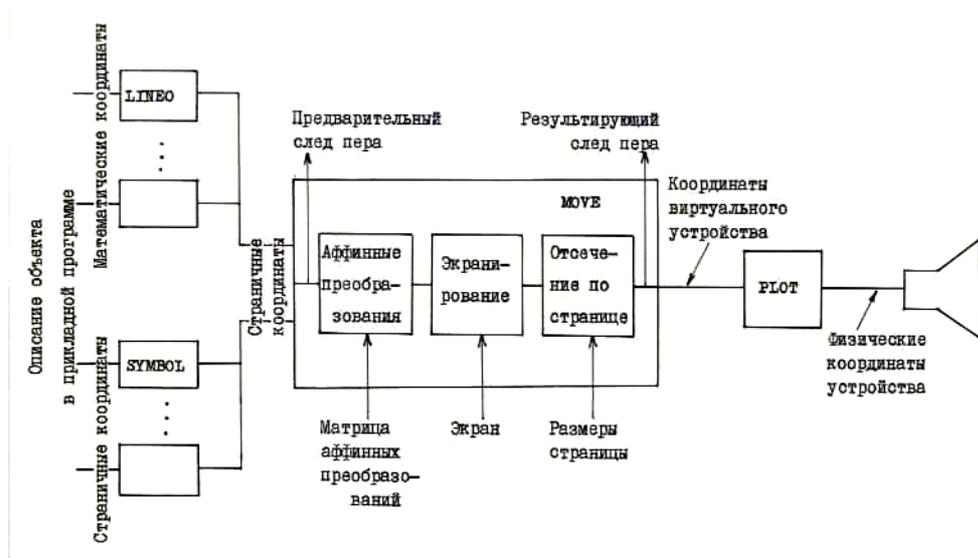


Рис. 1.1. Общая структура графической системы.

каком-либо другом носителе. Если для описания объекта применяются так называемые *математические координаты*, т.е. единицы измерения, свойственные модели (например, километры, миллибары, секунды и т.п.), то для получения графического образа необходимы *координаты устройства*, которые попадают в дисплейный файл или непосредственно используются *драйвером* для управления устройством.

Следовательно, графическая система должна обеспечить получение образа для заданного описания объекта с соответствующим преобразованием системы координат. Разумеется, один и тот же объект можно изобразить разными способами. Так, однозначной функции двух переменных может соответствовать плоская проекция поверхности, карта изолиний либо линий пересечения поверхности с заданными плоскостями и т.д. Выбор способа изображения - это, в сущности, выбор функции вывода в прикладной программе.

Кроме преобразования системы координат при переходе от объекта к его образу могут потребоваться и некоторые другие преобразования. В частности, в математическом пространстве можно определить прямоугольное окно и отсечь все части объекта, не попадающие в его пределы. Далее на видовой поверхности определяется прямоугольная область, занимающая, возможно, весь экран дисплея или, в случае графплоттера, всю страницу, и производится отображение окна на эту область.

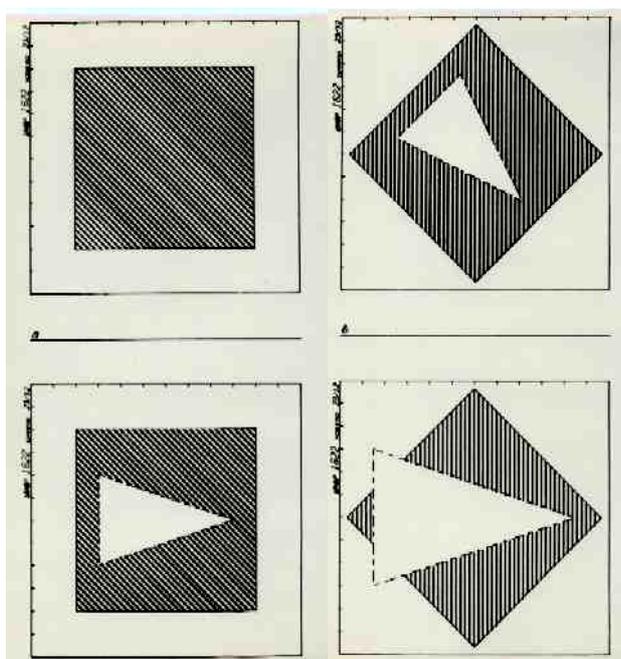


Рис. 1.2. Некоторые возможности формирования изображения:

- а) На странице задается квадратный участок, который штрихуется. Граница участка очерчивается сплошной линией.
- б) Задается треугольный экран. Граница экрана рисуется штрихпунктирной линией. Далее выполняются те же действия, что и в случае а).
- в) Задается преобразование поворота на 45° вокруг центра страницы (формируется матрица аффинных преобразований), и затем выполняются те же действия, что и в случае б).
- г) Устанавливается преобразование масштабирования и задается экран такой же, как в случае б). Рисуется граница экрана (в результате преобразования изменяется размер экрана, а также длины штрихов и пунктиров). Затем преобразование масштабирования отменяется. Устанавливается преобразование поворота на 45° и, наконец, выполняются те же действия, что и в случае а).

В процессе вывода к образу объекта можно применить необходимые аффинные (линейные) преобразования. Те части образа объекта, которые закрыты *экраном* или оказались за пределами страницы, на видовой поверхности не изображаются. Предусматривается возможность получения *следа пера* (т.е. фиксирования его траектории при рисовании) как предварительного (до преобразований), так и истинного. Информация о следе пера может использоваться, например, для выполнения нелинейных преобразований графического образа.

В итоге, общую структуру графической системы можно представлять себе такой, как она изображена на рис. 1.1. В дальнейшем мы ограничимся, в основном, средствами вывода и преобразования рисунка, имея в виду их конкретную реализацию в графическом пакете Графор. На рис. 1.2 показано как можно использовать при формировании изображения аффинные преобразования и экранирование.

1.2. Особенности реализации Графора, обеспечивающие его мобильность

В Графоре вся специфика конкретного графического устройства сосредоточена в программах PLOT и BLOCK DATA (или в *драйвере* и *программе инициализации*). Это следует учитывать при адаптации комплекса для новых графических устройств.

Все начальные установки в общих блоках вынесены в отдельную подпрограмму. На БЭСМ-6 - это BLOCK DATA.

BLOCK DATA

C..ТАБЛИЦА СВЯЗИ БАЗИСНЫХ ПРОГРАММ

COMMON/GFTAB/IRDB(5),RDB(17)

DATA IRDB/3*0, 2*6/,RDB/1000.,0.,60.,26.,12*0.,1000./

C..ВЫБОР ПЕРА

COMMON/GFCOL/NCOL,JPEN,JCOL,JCOLC,IDX,IDY,IR(2,3)

DATA NCOL/3/,JPEN/0/,JCOL/1/,JCOLC/0/,IDX,IDY/2*0/,IR/0,0,0,140,0,280/

C..АФФИННЫЕ ПРЕОБРАЗОВАНИЯ

COMMON/GFATRM/A(6),B(6),N

COMMON/GFATRNR/R(6),IS

DATA A/1., 3*0.,1.,0./,B/1., 3*0.,1.,0./, N/0/

DATA R/1., 3*0.,1.,0./, IS/0/

C..ЭКРАНИРОВАНИЕ

COMMON/GFBLAN/JJ,XX1,XX2,YY1,YY2,NCOP,JCH,KCH,XXX,YYY,ICH

DATA JJ,KCH,XX1,XX2,YY1,YY2/0,0,4*0./

C..СЛЕД ПЕРА

COMMON/GFNTCH/CHEK3(50),NSIGN

COMMON/GFGOBS/NJCH,NKCH,JG,NRIS,NACH(16)

DATA NSIGN,NKCH,JG,NRIS,NACH/0,1,0,0,16*8/

C..БУФЕР ХРАНЕНИЯ ГЕОМЕТРИЧЕСКИХ ЭЛЕМЕНТОВ

COMMON/GFGEL/GF(28)

DATA GF/8*0., 1.,18*0.,0.01745329/

C..ФИКСАЦИЯ ОШИБКИ ПРОГРАММОЙ GRAFER

COMMON/GFERR/NN,ITER(4)

DATA NN/0/

C..БЕРГ-ШТРИХИ В ИЗОЛИНИЯХ

COMMON/GFBET/KT,ISTEP,SIZE

DATA KT,ISTEP,SIZE/0,0,0./

END

В версии Графора для ЕС ЭВМ эти функции выполняет программа GRINIT. Связь в базисных программах осуществляется через общий блок GFTAB. Его состав, а также смысл содержащихся в этом блоке величин поясняются в табл. 1.

Функции, выполняемые программой PLOT, в значительной степени зависят от аппаратных возможностей графического устройства и операционной обстановки. Так при использовании простых устройств, не имеющих аппаратной генерации линий, алгоритм линейной интерполяции включается в саму программу PLOT.

В некоторых своих вариантах комплекс Графор непосредственно управляет графическим устройством. Это неизбежно при работе на машинах, где используется однопрограммный режим и отсутствуют прерывания от устройств. В больших машинах, которые работают под управлением мультипрограммной операционной системы, выдача команд графическому устройству выполняется супервизором операционной системы (соответствующая его часть называется *графическим супервизором*). К программе PLOT, которая взаимодействует с внешней средой, предъявляются дополнительные требования.

Так в случае использования комплекса Графор в рамках ОС ДИСПАК на машине БЭСМ-6 с шаговым графплоттером связь с операционной системой осуществляется с помощью экстракода. При реализации Графора на ЕС ЭВМ вследствие различия в системе приказов графических устройств и канальных программ ввода/вывода нижний уровень распадается еще на два: уровень генерации графических приказов (*компилирующая программа*) и уровень

собственно ввода/вывода (*хэндлер устройства*). Программа PLOT в этом случае будет выполнять функции посредника, обращаясь к соответствующим входам компилирующей программы (см. приложение А).

Привязка конкретного устройства к графическим программам верхнего уровня осуществляется с помощью программы GRINIT.

Т а б л и ц а 1

Область связи

COMMON /GFTAB/IRDB(5),RDB(17)		
Элемент общего блока GFTAB	Смысл	Программа, изменяющая значение
IRDB(1)	Номер набора литер	SET
IRDB(2)	Признак страницы	PAGE
IRDB(3)	Зависит от реализации	
IRDB(4)	Количество байт для представления целого числа с плавающей точкой (для БЭСМ-6 – число 6)	
IRDB(5)	Количество байт для представления числа с плавающей точкой (для БЭСМ-6 - число 6)	
RDB(1)	Количество шагов виртуального устройства в выбранной единице измерения (CMS, MMS, INCHES), первоначально - 1000	
RDB(2)	Наклон литеры	ITALIC
RDB(3)	Максимальные размеры страницы по оси X	
RDB(4)	Максимальные размеры страницы по оси Y	
RDB(5)	Длина стороны страницы по X	PAGE
RDB(6)	Длина стороны страницы по Y	PAGE
RDB(7), RDB(8)	XOR, YOR - координаты левого нижнего угла прямоугольной области (REGION) или полярной области (POLREG)	REGION POLREG
RDB(9), RDB(10)	XLR, YLR - длины сторон прямоугольной области (REGION) или внутренний и внешний радиусы полярной области (POLREG)	REGION POLREG
RDB(11), RDB(12), RDB(13), RDB(14)	XMIN, YMIN, XMAX, YMAX - пределы изменения функции и аргумента	LIMITS
RDB(15)	THO - начальный угол полярной области	POLREG
RDB(16)	THF - конечный угол полярной области	POLREG
RDB(17)	Число 1000 – характеристика виртуального устройства (количество шагов в сантиметре), т.е шаг виртуального устройства выбран равным 0.01 мм	

Вызов этой программы необходимо выполнить до обращения к каким-либо программам Графора. Завершает вывод программа GRFIN. К ней пользователь должен обратиться после окончания работы главной программы, в противном случае часть данных (или все) могут быть утеряны. Обе программы без параметров.

Программа GRINIT определяет тип выводного устройства. При этом в случае графического устройства производится загрузка соответствующей компилирующей программы и хэндлера устройства. Кроме того, устанавливается связь программы PLOT с компилирующей программой, а компилирующей программы с хэндлером устройства. После этого выполняются иницирующие функции компилирующей программы и хэндлера устройства.

Если же выходным устройством является перфолента или магнитная лента, то программа GRINIT выдает запрос оператору с требованием указать тип графического устройства, для которого предназначается подготавливаемая лента. После ответа оператора, как и в предыдущем случае, производится загрузка соответствующей компилирующей программы и модуля графического вывода на ленту. Использование другой компилирующей программы обусловлено тем, что для некоторых устройств (например, ЕС-7054) форматы данных в режиме ONLINE отличаются от форматов в режиме OFFLINE.

Реализация нижнего уровня для других машин может отличаться от описанного выше.

Для обеспечения графических устройств, имеющих несколько пишущих элементов, в Графоре имеется установочная программа SETPEN(J), работающая с общим блоком GFCOL. Параметр этой программы: J - номер пера. Структура блока GFCOL зависит от параметров устройства и программы, реализующей работу со сменой пишущих элементов. Например, в случае использования ЕС-7052 этот блок содержит следующие величины:

- 1 - число пишущих элементов,
- 2 - состояние опущено/поднято,
- 3 - требуемый номер,
- 4 - текущий номер,
- 5 - смещение по оси X при смене пера,
- 6 - смещение по оси Y при смене пера,
- 7 - двумерный массив расстояний от первого элемента в шагах виртуального устройства по двум осям.

Если же устройство поддерживает несколько различных *типов линий*, то для их установки может использоваться программа SLINST(J). Параметр J - номер типа линии: J = 1 - сплошная линия, J = 2 - штрихпунктирная линия, J = 3 - пунктирная линия.

В Графоре имеется еще несколько программ, зависящих от конкретной ЭВМ. Об этом нужно помнить при переносе комплекса на новые машины. Имена этих программ: SYMTAB, IDENT, IADR (функция), BCD, IBCD, RAND, MARKER.

1.3. Кадр, страница

Прежде всего необходимо отметить важное для дальнейшего изложения различие между дисплеем и графплоттером. Для каждого типа дисплея размер экрана и, следовательно, размер кадра постоянны и с ними жестко связана система координат устройства. Скажем, кадр на экране дисплея ЕС-7064 - это квадрат размером 250 мм×250 мм и по каждой координате адресуется 1024 точки. В этом случае удобно применять нормированные координаты устройства, изменяющиеся в пределах от 0 до 1. Тем самым достигается независимость от размера экрана. Правда, при таком подходе невозможно сохранить линейные размеры образов на дисплеях с разными размерами экранов. В графплоттерах, особенно в графплоттерах рулонного типа, размер кадра (применительно к графплоттерам мы будем называть его страницей) не фиксирован. Длина страницы может существенно (во много раз) отличаться от ее ширины. Кроме того, иногда (например, при подготовке фотошаблонов) должна сохраняться не только форма, но и абсолютные размеры элементов рисунка.

В Графоре *страница* - это прямоугольное поле на бумаге или экране дисплея, в пределах которого должны размещаться области для графиков, тексты и другие графические элементы. При задании страницы определяются ее размер, связанная с ней система координат и фиксируется *единица измерения*. Существуют три программы CMS, MMS и INCHES, устанавливающие в качестве единицы измерения сантиметр, миллиметр и дюйм соответственно. Эти программы не имеют параметров. Если ни одна из этих программ не вызывалась, то устанавливаются сантиметры). Единица измерения выбирается до определения страницы (размеры самой страницы указываются в этих единицах) и не может быть изменена внутри страницы. Таким образом, масштаб рисунка не зависит от шага конкретного графплоттера или размеров экрана дисплея.

Программа PAGE(XL, YL, NAME, N, J) позволяет определить страницу и имеет следующие параметры:

- XL, YL - размеры страницы вдоль осей X и Y;
- NAME - название страницы,

N - количество литер в названии;

J - признак очерчивания границы:

J = 0 - граница не очерчивается, J = 1 - граница очерчивается.

Определение страницы имеет принципиальное значение. Если страница не определена, то любые обращения к графическим программам игнорируются. При определении страницы считается, что перо графплоттера находится в левом нижнем углу страницы, т.е. в точке с координатами (0,0), тогда правый верхний угол страницы имеет координаты (XL, YL). Поэтому расположение первой страницы зависит от действий человека, включившего устройство и подготовившего его к работе. Расположение каждой следующей страницы выбирается автоматически программой ENDPG.

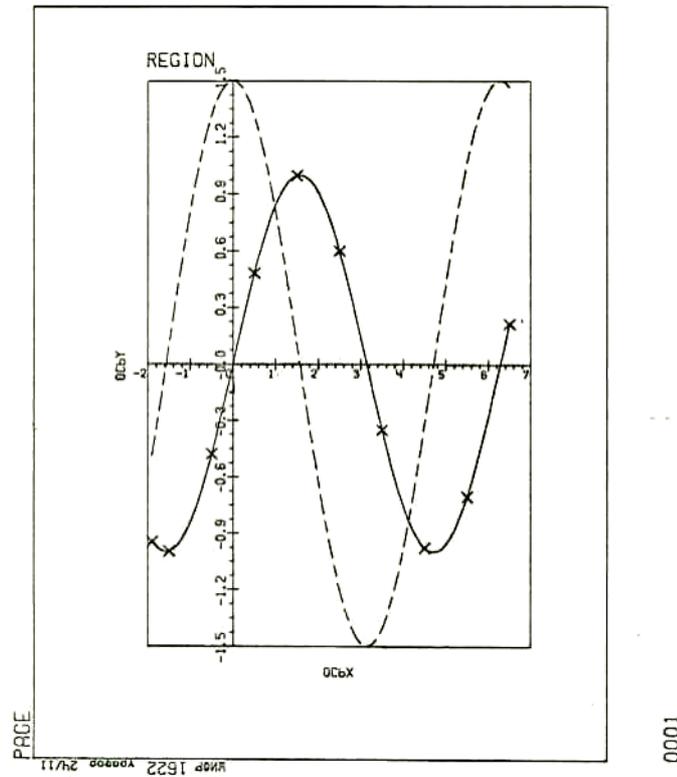


Рис. 1.3. Пример построения графика функции.

Существуют предельные, т.е. максимально допустимые размеры страницы, которые зависят от выбора графического устройства. Если какая-либо из координат выходит за указанные пределы, она полагается равной предельному значению. Перо в процессе рисования не может выйти за границы, установленные программой PAGE (даже если они не очерчены). Его можно вывести за пределы страницы только после того, как страница будет закрыта. Поэтому, если графический элемент или какая-то его часть лежит за пределами страницы, то либо весь элемент не рисуется, либо изображается только его часть, находящаяся в пределах страницы. Такая локализация пера внутри рабочего поля позволяет в случае ошибки сохранить в неприкосновенности ранее подготовленные страницы.

Название страницы пишется за ее пределами вдоль левой границы. Также за пределами страницы, вдоль нижней границы, пишутся имя (шифр) задачи и дата решения (рис.1.3 и 1.4). Эти данные извлекаются из операционной системы с помощью программы IDENT(MD,ND), которая формирует полученные из операционной системы имя (шифр) задачи MD и дату ND.

При любом перемещении пера из точки в точку запоминаются его текущие координаты как результирующие, так и предварительные.

Результирующие координаты пера - это те, которые получаются после применения всех преобразований (линейных, экранирования, отсечения по странице).

Предварительные координаты - это координаты до выполнения преобразований.

Узнать значения этих координат можно с помощью подпрограмм WHERE и WHERP.

Программа WHERE(X,Y,F) позволяет узнать результирующие координаты пера на странице (параметры X, Y), а также коэффициент F перевода сантиметра в выбранные единицы измерения.

Программа WHERP(X,Y,F) аналогична предыдущей, но позволяет определить предварительные координаты текущего положения пера (параметры X, Y) и коэффициент F перевода сантиметра в выбранные единицы измерения.

После окончания вывода на текущую страницу ее необходимо закрыть, только после этого можно определить следующую страницу.

```

Тест    ГРАФОР

DIMENSION X(100),Y(100),Z(100)
X(1)=-1.9
Y(1)=SINI(-1.9)
Z(1)=COSI(-1.9)*1.5
DO 5,I=2,90
X(I)=X(I-1)*.1
Y(I)=SINI(X(I))
5 Z(I)=COSI(X(I))*1.5
CALL PAGE(15.,20.,4,PAGE,4,1)
CALL LIMITS(-2.,7.,-1.5,1.5)
CALL REGION(3.,3.,10.,15.,6,REGION,6,1)
CALL AXES(ЧНОСЬХ,4.,1.,5.,ЧНОСЬУ,4.,3.,4.,0)
CALL LINEMO(X,Y,05,2,10)
CALL BROKEN(.5.,2.,3.,2)
CALL BRLINE(X,Z,05)
CALL ENDPG(ЧНО001)
END

```

0002

ИИР 1622 графоп 24/11

Рис. 1.4. Программа, с помощью которой выполнено построение графика на рис. 1.3.

Программа ENDPG(NUMB) закрывает доступ к текущей странице и выполняет начальные установки, необходимые для определения следующей страницы. При этом в случае графопостроителя перо выводится за пределы страницы, проводится линия разреза между страницами параллельно оси Y на расстоянии 2 см от правой границы закрываемой страницы. Вдоль линии разреза пишется текст, заданный параметром NUMB, затем перо отводится от линии разреза вправо на расстояние 2 см в точку, которая станет левым нижним углом новой страницы. После того как страница закрыта, может быть изменена и единица измерения.

Обращение к программе ENDPG обязательно не только между текущими страницами данной задачи, но и после последней страницы, чтобы тем самым исключить наложение на эту страницу рисунков другой задачи.

В Графоре имеется также программа NEWFRM, которая предназначена для смены кадра при выводе изображения на дисплей или планшетный графоплоттер. Обращение к этой программе приостанавливает работу основной программы. Продолжить ее можно лишь после вмешательства оператора. Например, на ЕС ЭВМ с дисплеем ЕС-7064 для этого необходимо нажать функциональную клавишу. При этом очищается память дисплея и возобновляется вывод.

1.4. Прямая линия

В графическом пакете каждому объекту соответствует подпрограмма (функция вывода), интерпретирующая описание объекта и генерирующая его образ как композицию других более простых объектов. Можно выделить совокупность некоторых базисных объектов. К ним относятся: прямая линия, алфавитно-цифровые и специальные знаки, маркеры, правильные многоугольники, прямоугольники, дуги окружности и эллипса и др. Но примитивным (неразложимым) является только один объект - *отрезок прямой линии*.

В Графоре роль генератора прямых линий выполняет программа MOVE. Эта программа позволяет перевести перо из точки, в которой оно находится в текущий момент, в любую другую точку внутри страницы. При этом перо может перемещаться, оставляя след на бумаге, если оно опущено, или не оставляя следа, если оно поднято. Заметим, что точкой отсчета считается точка с нулевыми координатами в левом нижнем углу страницы. Поэтому координаты всех других точек задаются относительно этого угла в тех единицах измерения, которые были выбраны перед определением страницы. Программа MOVE выполняет преобразование заданных координат в шаги виртуального графплоттера. Непосредственно же управляет движениями пера программа PLOT.

Назначение программы PLOT состоит в том, чтобы отобразить виртуальное устройство на то или иное конкретное устройство. В ней учитываются все специфические особенности используемого устройства. Поэтому при смене устройства эта программа подвергается изменениям. Это существенно, поскольку при наличии весьма разнообразных графических устройств графический пакет должен быть относительно независим от устройств и от способа их подключения. К программе PLOT не следует обращаться из прикладной программы.

Программа PLOT(*IX, IY, J*) позволяет присвоить указанные значения координат точке, в которой в данный момент находится перо, или перевести перо в указанную точку с вычерчиванием или без вычерчивания прямой линии. Параметры программы:

IX, IY - целочисленные значения координат *X* и *Y*, заданные в шагах (размер шага равен 0.01 мм);

J - управление пером:

J = 0 - присвоить значения координат точке, в которой находится перо,

J = -2 - перевести перо из текущей точки в точку с координатами (*IX, IY*) без вычерчивания линии,

J = 2 - перевести перо из текущей точки в точку с координатами (*IX, IY*) с вычерчиванием прямой линии.

Программа MOVE(*X, Y, J*) позволяет переместить перо из текущей точки в точку с заданными координатами без рисования, провести прямую от текущей точки к точке с заданными координатами или присвоить заданные значения точке, в которой находится перо. Параметры программы:

X, Y - координаты точки, в которую перемещается перо;

J - признак вычерчивания линии: *J* = 0 или 990 - перемещение без рисования,

J = 1 или 991 - перемещение с рисованием,

J = 992 - присвоить текущей точке значения координат *X, Y*.

Заметим, что если *J* = 0 или *J* = 1 и значение какой-либо из координат находится за пределами страницы, то этой координате присваивается граничное значение. При *J* = 990, 991 такой контроль не производится и разрешается рисование за пределами страницы. *J* = 990, 991, 992 - допускаются только в программах Графора.

Таким образом, в результате срезки по границе вблизи нее могут возникнуть некоторые искажения рисунка. Такое отсечение, однако, носит чисто технический характер. Для точного отсечения следует воспользоваться программой экранирования.

Существует еще несколько вариантов программы MOVE, учитывающих режимы работы с экранированием (MOVE1, MOVE3) и формированием следа пера (MOVE2, MOVE3). Подробнее об этом сказано в параграфах 2.2 и 2.3.

Итак, программа MOVE(*X, Y, J*) строит отрезок по координатам конечной его точки. В качестве начальной точки отрезка берется текущая точка, т.е. та, в которой оказалось перо перед обращением к соответствующей программе.

Как вы уже заметили, при описании программ термином перо обозначается не только перо графплоттера, но и пишущий элемент любого графического устройства, например электронный луч ЭЛТ.

Если при перемещении пишущего элемента остается след на видовой поверхности, то мы будем говорить, что *перо опущено*, в противном случае - *перо поднято*.

В Графоре имеется еще несколько программ для перемещения пера по прямой в опущенном или поднятом состоянии.

Программа MOVA(DL,TH,J) строит отрезок по его длине DL и углу с осью x. Угол TH задается в градусах (рис. 1.5, а).

Программа MOVV(DX,DY,J) строит отрезок по приращениям координат DX и DY вдоль осей x и y соответственно (рис. 1.5, б).

Программа MOVC(XM,YM,DL,J) строит отрезок по его длине DL и точке (XM, YM), лежащей на отрезке или его продолжении. Если $DL > 0$, то перо из начальной точки движется в сторону точки (XM, YM), в противном случае - в противоположную. Точка (XM, YM) может находиться вне страницы (рис. 1.5, в).

Для этих программ: J = 0 - перо поднято, J = 1 - перо опущено.

В рисунках и технических чертежах используются, как правило, различные типы линий: сплошные, штриховые, штрихпунктирные, линии различной толщины. Программы для построения таких линий также описаны в этом разделе.

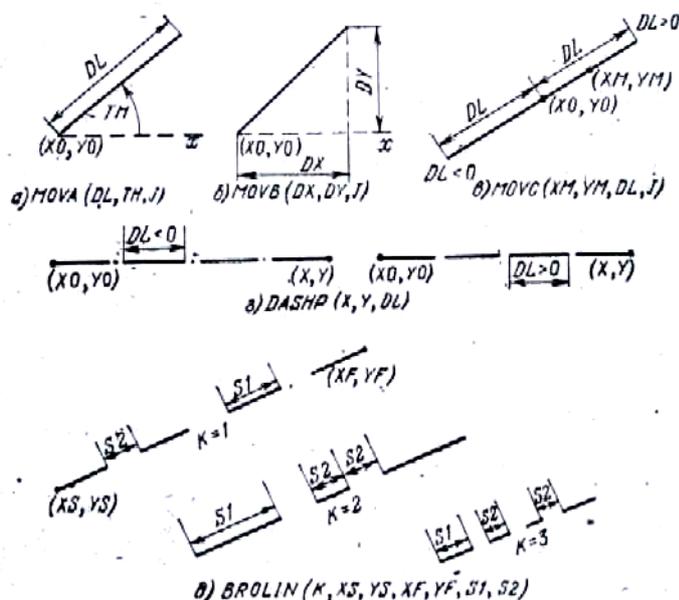


Рис. 1.5. Типы прямых линий.

Программа FATLIN(X,Y,D) позволяет провести из текущей точки в заданную точку прямую линию указанной толщины. Ее параметры:

- X, Y - координаты конечной точки,
- D - толщина линии в миллиметрах.

Поскольку толщина линии достигается лишь трехкратным проведением, при больших D линия может оказаться не налитой.

Программа DASHP(X,Y,DL) позволяет провести из текущей точки в заданную точку пунктирную или штрихпунктирную прямую линию (рис. 1.5, з). Параметры программы:

X, Y - координаты точки, в которую проводится линия;

|DL| - длина основного штриха линии: $DL > 0$ - проводится штриховая линия, $DL < 0$ - проводится штрихпунктирная линия.

В штриховой линии расстояние между штрихами (интервал) равно $0.3 * |DL|$. В штрихпунктирной линии длина промежутка между штрихами равна $0.23 * |DL|$ и в середине этого промежутка изображается дополнительный штрих размером $0.03 * |DL|$.

Программа BROLIN(K,XS,YS,XF,YF,S1,S2) позволяет провести три типа штриховых линий (рис. 1.5, д). Параметры программы:

К - тип линии: К = 1 - штриховая линия, К = 2 - штрихпунктирная линия, К = 3 - штрихпунктирпунктирная линия;
XS,YS - координаты начальной точки;
XF,YF - координаты конечной точки;
S1 - длина штриха;
S2 - длина пунктира и интервала.

1.5. Тексты, числа, маркеры

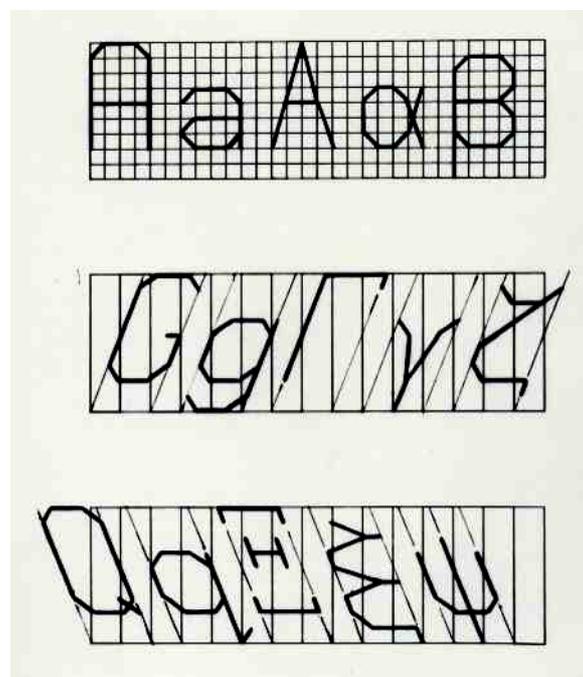
1.5.1. Вывод текстовой информации. В некоторых устройствах имеются аппаратные генераторы символов. Но они обладают довольно ограниченными возможностями в отношении выбора шрифтов, размеров литер и углов наклона текстовых строк. Кроме того, во многих устройствах, особенно в графплоттерах, такие генераторы вообще отсутствуют.

При программной реализации генератора символов предполагается, что каждая литера кодируется как последовательность штрихов. Следовательно, необходимо иметь таблицу с описанием каждой литеры. При большом количестве литер (в Графоре, например, их 176) проблемы компактного описания литеры и экономной организации таблицы становятся весьма серьезными.

В Графоре используется известный и хорошо зарекомендовавший себя способ кодирования образа литеры, при котором каждая литера описывается последовательностью не более, чем шестнадцати штрихов на сетке размером 4×9 (см. рис. 1.6).

Ширина литеры составляет $4/7$ высоты прописной буквы, расстояние между двумя соседними литерами равно половине ширины литеры.

Генератор символов состоит из двух программ. Программа SYMTAB, написанная на автокоде, содержит *таблицу кодов литер*. При вызове этой программы указываются слово и номер литеры в слове, а также номер набора (все множество литер разделено на 4 набора, текущим считается тот набор, который был задан при вызове программы SET). Результатом работы программы SYMTAB является массив пар координат для последовательности штрихов, воспроизводящих указанную литеру, и величина, равная фактическому количеству штрихов.



```
CALL ITALIC( 1 )  
CALL ITALIC( 0 )  
CALL ITALIC( -1 )
```

Рис. 1.6. Примеры изображения литер

Программа SYMTAB(JTEXT,I,NPLOT,KK,ISET) позволяет выбрать из таблицы массив координат для построения литеры. Параметры программы:

JTEXT - слово текста;

I - номер литеры в слове;

NPLOT - массив координат для построения литеры;

KK - количество пар координат, задающих литеру;

ISET - номер набора.

Основная же программа генератора символов SYMBOL написана на фортране. Эта программа пишет строку текста заданной длины под заданным углом к оси X. Строку можно начать либо в указанной точке, либо указать ее место относительно конца предыдущей строки.

При работе программы SYMBOL строка текста выводится на графплоттер последовательно, литера за литерой, причем, если координаты очередной литеры оказываются за пределами страницы, то она не рисуется. Таким образом, выводится только та часть строки текста, которая размещается в пределах страницы.

Программа SYMBOL(X,Y,SIZE,JTEXT,N,THETA) позволяет написать заданный текст. Параметры программы следующие:

X, Y - координаты левого нижнего угла первой литеры текста или приращения к конечным координатам текста, предшествующего данному;

SIZE - высота прямоугольника, в котором вычерчивается литера;

JTEXT - заданный текст;

|N| - количество литер в строке: N > 0 - (X, Y) считаются координатами на странице, N < 0 - (X, Y) считаются приращениями;

THETA - угол наклона строки текста к оси X (в градусах).

Программа SET(J) позволяет выбрать один из четырех наборов литер (рис. 1.7). Имеются наборы, в которые входят:

1) прописные русские и латинские буквы, цифры и знаки,

2) строчные русские и латинские буквы, цифры и знаки,

3) прописные греческие буквы, цифры, знаки и спецсимволы,

4) строчные греческие буквы, цифры, знаки и спецсимволы.

Цифры и знаки во всех наборах одинаковы.

Обращаться к программе SET можно как до, так и после определения страницы. Информация о выбранном наборе будет сохраняться до следующего обращения к программе SET в пределах одной страницы. Если не было ни одного обращения к программе SET, то используется набор, содержащий прописные русские и латинские буквы. Этот набор восстанавливается и при закрытии каждой страницы.

```
SET( 0 )  /+- . )$*( = ' : БГДЖЗИЙЛПФЦЩЩЬЬЪЭЮЯАВСDEF GHI JKLMNOPQRSTUVWXYZ0123456789

SET( 1 )  /+- . )$*( = ' : бгджзюилпфцщщььэюяабсdef gni jklmnopqrstuvwxyz0123456789

SET( 2 )  /+- . )$*( = ' : ПУСД <=> V * > z < s & | ≡ ~ ± А В Г Δ Ε Φ Η Ι Θ Κ Λ Μ Ν Ο Π Ε ρ Σ Τ Υ Φ Ω Χ Υ Ζ 0 1 2 3 4 5 6 7 8 9

SET( 3 )  /+- . )$*( = ' : ∫ √ < > ∥ ⊥ [ ] † ‡ ← → ∂ ∇ ! ? @ \ % α β γ δ ε φ γ η ι θ κ λ μ ν ο π ρ σ τ υ φ ω χ υ ζ 0 1 2 3 4 5 6 7 8 9
```

Рис. 1.7. Наборы литер

Кроме обычного прямого шрифта можно получить и *курсив*, установив соответствующий режим обращением к программе ITALIC(J). Ее параметр J задает признак курсива: J = 0 - прямой шрифт, J = 1 - правый наклон, J = -1 - левый наклон. Тогда угол наклона каждого

символа к строке текста будет составлять около 70° или 110° (наклон может быть соответственно вправо или влево).

Обращаться к программе ITALIC можно любое число раз как до, так и после определения страницы. Установленный режим наклона символов распространяется на все тексты до следующего обращения к программе ITALIC в пределах одной страницы. Если не было ни одного обращения к этой программе, действует стандартный режим - прямой шрифт. Этот же стандартный режим устанавливается при закрытии страницы. Примеры работы с программой ITALIC приведены на рис. 1.6.

1.5.2. Вывод чисел. В Графоре роль генератора чисел выполняет программа NUMBER. Она дает возможность написать на странице число, представленное в машине в нормализованном двоичном виде. Форма обращения к программе NUMBER во многих отношениях аналогична форме обращения к программе SYMBOL. Программа NUMBER переводит числа в текстовый вид и затем с помощью программы SYMBOL рисует их.

Число может состоять из знака числа, целой части, точки и дробной части. Если число положительно, то знак опускается. Количество дробных знаков задается в обращении к программе. Если оно равно 0, то точка не пишется.

Программа NUMBER(X, Y, HGF, FNUM, N, TH) дает возможность перевести и написать число. Параметры программы:

X, Y - координаты точки, в которой надо начать писать число, или приращения к конечным координатам текста, предшествующего данному;

|HGF| - высота прямоугольника, в котором вычерчивается цифра:

HGF > 0 - (X, Y) считаются координатами на странице,

HGF < 0 (X, Y) считаются приращениями;

FNUM - заданное вещественное число;

N - количество требуемых дробных знаков (если $N < 0$, то пишутся старшие цифры целой части, N младших цифр целой части отбрасываются).

TH - угол наклона строки текста к оси x (в градусах).

Имеются, кроме того, две программы (BCD, IBCD) для перевода чисел из внутренней машинно-зависимой формы в текстовый вид, приемлемый для программы SYMBOL.

Программа BCD(FNUM, JT, N) предназначена для перевода вещественных чисел, а программа IBCD(NUM, JT) - для перевода целых чисел. Здесь FNUM, NUM - задаваемые числа, JT - переведенное число в текстовом виде, N - число дробных знаков.

Однако, эти программы обладают несколько ограниченными возможностями. Они рассчитаны на то, что выводимое число занимает не более 6 текстовых позиций (с учетом знака и точки, если они требуются). Поэтому, если не помещается дробная часть, то выдается столько знаков после точки, сколько можно поместить. Если не помещаются целая часть и точка, то вместо переведенного числа появляется текст `_____ * *`. Число, которое занимает меньше шести позиций, дополняется пробелами слева.

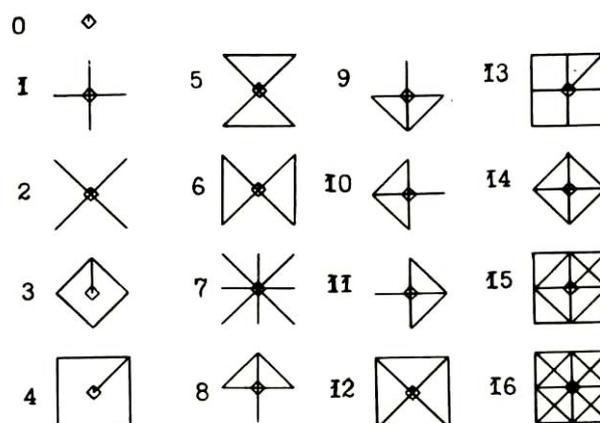


Рис. 1.8. Маркеры

Существует несколько реализаций этих программ как на автокоде, так и на Фортране. В последнем варианте используется служебная программа BCDX. В зависимости от

используемой ЭВМ, переведенное число может помещаться в одном машинном слове (БЭСМ-6) или занимать первые 6 байтов в массиве из нескольких слов (ЕС ЭВМ).

Программы BCD и IBCD являются служебными для некоторых программ Графора.

1.5.3. Вывод маркеров. Маркеры предназначены для выделения точек. Там, где находилось перо в момент обращения к программе MARKER, рисуется точка, затем вокруг этой точки рисуется маркер. По окончании рисования перо выводится в позицию, откуда начиналось рисование. Величина маркера такова, что он вписывается в квадрат размером 3×3 мм. Задавая при обращении к программе номер маркера отрицательным, можно нарисовать маркер вдвое меньшего размера. В Графоре существует 17 различных маркеров. Они показаны на рис. 1.8 (в увеличенном размере). Сложные маркеры строятся как суперпозиции простых. Точка рисуется как маленький ромбик (маркер № 3). Программа MARKER(NMARK) предназначена для вычерчивания маркеров. Здесь NMARK - номер маркера; $|NMARK| \leq 16$. Если $NMARK < 0$, рисуется маркер вдвое меньшего размера.

1.6. Графические элементы и геометрические фигуры

В этом разделе описаны программы для построения различных геометрических элементов и фигур. Их линейные размеры задаются в предварительно выбранных единицах измерения.

1.6.1. Многоугольники. Программа TRIGL($X_0, Y_0, W, H, THETA, SLOPE$) рисует треугольник с заданными основанием, высотой и углом между основанием и одной из сторон (рис. 1.9, а). Параметры программы:

X_0, Y_0 - координаты начальной точки;

W - длина основания;

H - высота треугольника;

$THETA$ - угол, образованный пересечением двух сторон (в градусах);

$SLOPE$ - угол наклона основания к оси x (в градусах).

Программа BOX(X, Y, XL, YL) позволяет начертить прямоугольник заданных размеров со сторонами, параллельными осям. Перо выводится в точку (X, Y) и далее чертится прямоугольник в направлении по часовой стрелке (рис. 1.9, б). Параметры программы:

X, Y - координаты левого нижнего угла прямоугольника;

XL - размер стороны, параллельной оси x ;

YL - размер стороны, параллельной оси y .

Программа RECT(X, Y, H, W, TH) позволяет начертить прямоугольник заданных размеров, повернутый на заданный угол. Прямоугольник чертится из точки (X, Y) в направлении по часовой стрелке (рис. 1.9, в). Параметры программы:

X, Y - координаты левого нижнего угла прямоугольника;

H, W - длины сторон прямоугольника;

TH - угол поворота вокруг точки (X, Y) относительно оси x (в градусах).

Программа GRID(X_0, Y_0, XS, YS, M, N) строит равномерную прямоугольную сетку со сторонами, параллельными осям координат (рис. 1.9, г). Ее параметры:

X_0, Y_0 - координаты левой нижней вершины прямоугольника;

XS, YS - шаги разбиения сетки по ширине и высоте;

M - количество шагов по ширине сетки;

N - количество шагов по высоте сетки.

Программа ANGRID($X_0, Y_0, XS, YS, M, N, SLOPE$) позволяет нарисовать прямоугольную сетку с поворотом на заданный угол. Параметры программы аналогичны соответствующим параметрам программы GRID.

Параметр $SLOPE$ задает угол наклона основания сетки к оси x (рис. 1.9, г, д).

Программа POLYG(X, Y, SL, M, TH) позволяет начертить правильный многоугольник (рис. 1.9, е). Параметры программы:

X, Y - координаты начальной точки;

SL - размер стороны многоугольника;

$|M|$ - число сторон многоугольника: $M > 0$ - рисуется выпуклый многоугольник, $M < 0$ - рисуется звездчатый многоугольник;

TH - угол наклона к оси x стороны, с которой начинается рисование (в градусах).

Программа POLG(R,M,PHI) изображает выпуклый правильный многоугольник с центром в текущей точке (рис. 1.9, ж). Ее параметры:

|R| - радиус описанной окружности или длина стороны: $R > 0$ - задан радиус описанной окружности, $R < 0$ - задана сторона многоугольника;

M - число сторон многоугольника;

PHI - угол от оси x до ближайшего луча, проведенного в вершину (в градусах).

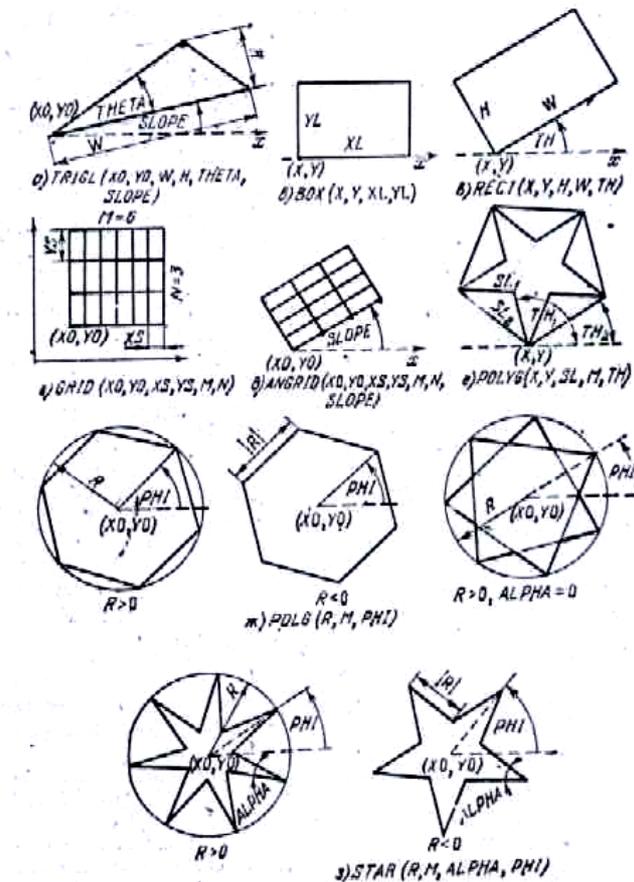


Рис. 1.9. Многоугольники

Программа STAR(R,M,ALPHA,PHI) рисует правильный звездчатый многоугольник с центром в текущей точке (рис. 1.9, з). Ее параметры:

|R| - радиус описанной окружности или длина стороны: $R > 0$ - задан радиус описанной окружности, $R < 0$ - задана сторона многоугольника;

M - число лучей в многоугольнике;

ALPHA - угол раствора луча в градусах: $ALPHA = 0$ - правильная звезда,

$ALPHA \geq 180^\circ * (M - 2) / M$ - правильный выпуклый M-угольник;

PHI - угол между осью x и ближайшим лучом, измеряемый против часовой стрелки.

1.6.2. Окружности и их дуги, спирали, сектор кругового кольца. Прежде чем перейти к описанию программ построения дуг окружностей, отметим общую особенность ряда программ, описываемых в данной главе. Если соотношение фактических параметров при обращении в какой-либо из программ геометрически не позволяет выполнять указанную операцию (например, расстояние между начальной и конечной точками дуги больше диаметра), то такое обращение мы будем называть *некорректным*. При некорректном обращении на печатающее устройство выдается диагностический текст НЕ ТЕ ПАРАМЕТРЫ, после чего происходит выход из данной программы без выполнения каких-либо действий. При этом, если в программе некоторым переменным должны быть присвоены какие-то значения, то такого присваивания тоже не произойдет. Об этом следует помнить, если значения этих переменных используются в последующих программах. От некорректного обращения следует отличать обращение с параметрами, значения которых лежат вне допустимой области определения (например,

отрицательные числа для страничных координат). В последних случаях никакой диагностической печати не будет, а результаты работы программ не определены.

Программа ARC(X1,Y1,X2,Y2,X3,Y3,J) позволяет провести окружность или дугу окружности по трем заданным точкам. Параметры программы:

X1, Y1; X2, Y2; X3, Y3 - координаты трех точек;

J - признак вычерчивания дуги: J = 1 - проводится дуга от первой точки к третьей через вторую, J = 0 - проводится полная окружность, J = -1 - от первой точки к третьей проводится дуга, не проходящая через вторую точку.

Программа CIRC(R) позволяет начертить окружность заданного радиуса с центром в текущей точке. После вычерчивания окружности перо возвращается в исходную точку. Параметр программы: R - радиус окружности.

Программа ARCIA(R,THS,THF) проводит из текущей точки дугу окружности с заданным наклоном начального и конечного радиусов (рис. 1.10, а). Параметры программы:

|R| - величина радиуса: R > 0 - перемещение пера из начальной точки в конечную против часовой стрелки, R < 0 - перемещение пера по часовой стрелке;

THS, THF - углы наклона начального и конечного радиусов к оси x (в градусах).

Если |THS-THF| > 360°, обращение к программе считается некорректным, хотя геометрически построение такой дуги возможно.

Программа ARCIB(R,XF,YF,J) соединяет текущую точку с заданной точкой дугой окружности заданного радиуса. Параметры программы:

|R| - величина радиуса: R > 0 - перемещение пера из начальной точки в конечную против часовой стрелки, R < 0 - перемещение пера по часовой стрелке;

XF, YF - координаты конечной точки;

J - признак выбора дуги: J = 0 - короткая дуга ($\leq 180^\circ$), J = 1 - длинная дуга ($\geq 180^\circ$).

Если расстояние между концами дуги равно диаметру окружности, то оба варианта дуг, соответствующие разным значениям J, совпадают. Если начальная точка совпадает с конечной, то для любого знака параметра R вычерчивается полная окружность с углами наклона начального и конечного радиусов соответственно 0° и 360°. Если расстояние между заданными концами дуги превышает величину диаметра, то обращение к программе считается некорректным.

Программа ARCIC(XM,YM,XF,YF,J) проводит в заданную точку из текущей точки дугу окружности, проходящей через дополнительно указанную точку. Параметры программы:

XM, YM - координаты дополнительной точки дуги;

XF, YF - координаты конечной точки дуги;

J - признак включения дополнительной точки: J = 0 - дуга проходит через дополнительную точку, J = 1 - дуга не проходит через дополнительную точку.

Дополнительная точка может находиться и вне страницы в тех случаях, когда дуга через нее не проходит. Если начальная точка совпадает с конечной, то вычерчивается полная окружность с диаметром, равным расстоянию между начальной и дополнительной точкой. Если все три точки лежат на одной прямой, то обращение к программе считается некорректным. Программа ARCIC использует служебную программу ARCC1.

Программа ARCID(XC,YC,PHI) из текущей точки для заданных координат центра окружности (XC, YC) проводит дугу заданной угловой величины (PHI) (рис. 1.10, б). Значение угла задается в градусах. Положительным значениям угла соответствует направление против часовой стрелки. Если |PHI| > 360°, то обращение к программе считается некорректным.

Программа FATARC(R,XF,YF,J,D) аналогична программе ARCIB, но проводит дугу утолщенной линией. Параметры R, XF, YF, J имеют тот же смысл, что и одноименные параметры программы ARCIB. Параметр D задает (как и в программе FATLIN) толщину линии в миллиметрах.

Программа CIRCLE(XS,YS,THS,THF,RS,RF,L) предназначена для вычерчивания окружностей, спиралей, дуг окружностей и дуг спиралей (рис. 1.10, в). Параметры программы:

XS, YS - координаты начальной точки;

THS, THF - углы наклона начального и конечного радиусов к оси x (в градусах);

RS - начальный радиус;

RF - конечный радиус;

L - признак непрерывности линии: L = 0 - штриховая линия, L = 1 - сплошная линия.

Если начальный и конечный радиусы равны, вычерчиваются окружности и дуги окружностей. При $R_F < R_S$ получаем скручивающуюся спираль, а при $R_F > R_S$ - раскручивающуюся.

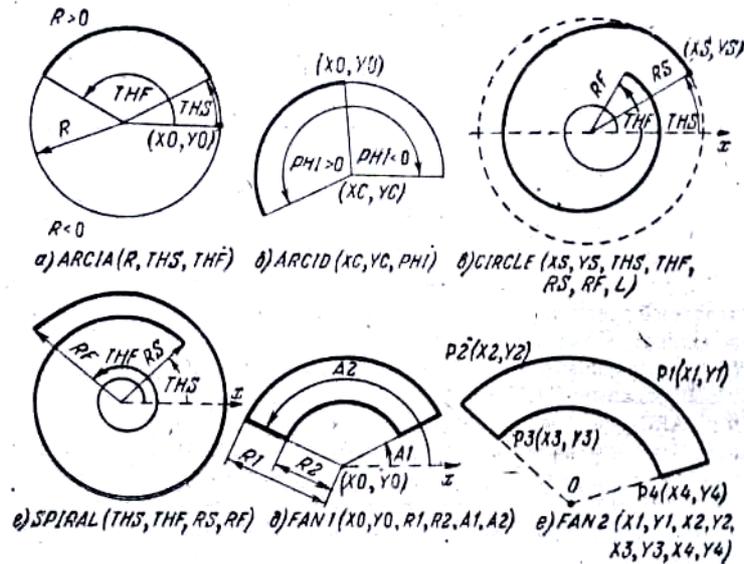


Рис. 1.10. Окружности, дуги окружностей и спирали.

Программа SPIRAL(THS, THF, RS, RF) позволяет начертить спираль с началом в текущей точке (рис. 1.10, *д*). Параметры программы:

THS, THF - углы наклона начального и конечного радиусов к оси x (в градусах);

RS, RF - начальный и конечный радиусы. Если $R_F > R_S$, рисуется раскручивающаяся спираль, а если $R_F < R_S$, то скручивающаяся. Если $R_F = R_S$, то рисуется дуга окружности.

Программа FAN1(X0, Y0, R1, R2, A1, A2) позволяет начертить сектор кругового кольца ("веера") (рис. 1.10, *д*). Параметры программы:

X0, Y0 - координаты центра окружностей;

R1 - радиус внешней дуги;

R2 - радиус внутренней дуги;

A1, A2 - углы наклона сторон сектора к оси x .

Программа FAN2(X1, Y1, X2, Y2, X3, Y3, X4, Y4) позволяет начертить сектор кругового кольца по заданным значениям координат четырех вершин (рис. 1.10, *е*). Эти вершины должны быть расположены так, чтобы $P_1P_4 = P_2P_3$ и $P_4O = P_3O$.

Программы FAN1 и FAN2 используют служебную программу ARC1, которая проводит в указанном направлении дугу окружности по заданным координатам центра, начальной и конечной точек.

1.6.3. Эллипсы, дуги эллипсов. Для вычерчивания эллипсов и их дуг в Графоре имеется четыре программы.

Программа ELIPS(XS, YS, A, B, ALPHA, THS, THF) позволяет начертить эллипс или дугу эллипса (рис. 1.11, *а*). Параметры программы:

XS, YS - координаты начальной точки;

A, B - размеры полуосей;

ALPHA - угол наклона полуоси A эллипса к оси x ;

THS - угол между начальным диаметром и осью A ;

THF - угол между конечным диаметром и осью A .

При равных осях получается окружность или ее дуга. Углы задаются в градусах.

Программа ELPS(A, B, ALPHA) позволяет начертить эллипс с заданными размерами полуосей с центром в текущей точке и с заданным наклоном полуоси A к оси x (рис. 1.11, *б*). Параметры программы:

A, B - размеры полуосей;

ALPHA - угол наклона полуоси A эллипса к оси x .

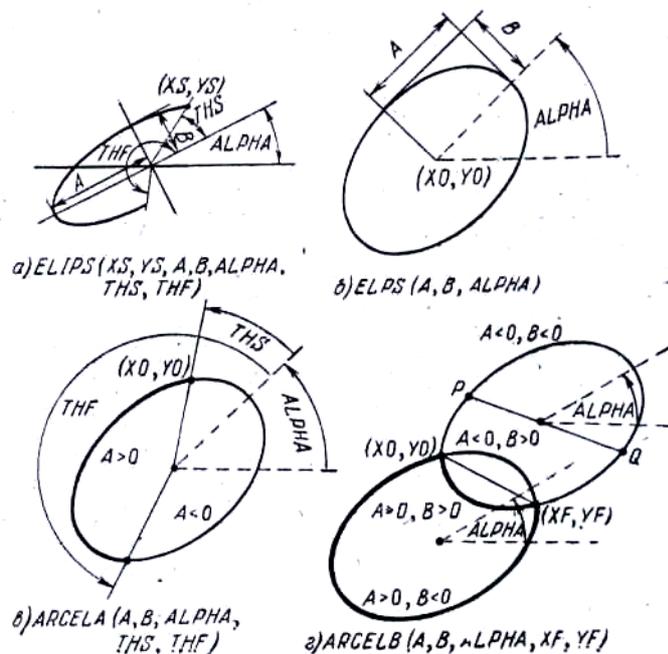


Рис. 1.11. Эллипсы и их дуги

Программа $ARCELA(A, B, ALPHA, THS, THF)$ позволяет провести из текущей точки дугу эллипса (рис. 1.11, в). Параметры программы:

$|A|$ - размер большой полуоси: $A > 0$ - перемещение пера против часовой стрелки, $A < 0$ - перемещение пера по часовой стрелке;

B - размер малой полуоси;

$ALPHA$ - угол наклона полуоси A эллипса к оси x (в градусах);

THS, THF - углы наклона начального и конечного радиуса к полуоси A .

Обращение к программе считается некорректным, если $|THS - THF| > 360^\circ$.

Программа $ARCELB(A, B, ALPHA, XF, YF)$ проводит из текущей точки в заданную точку дугу эллипса с указанными размерами полуосей и направлением рисования относительно центра (рис. 1.11, г). Ее параметры:

$|A|, |B|$ - размеры полуосей: $A > 0$ - перемещение пера против часовой стрелки, $A < 0$ - перемещение пера по часовой стрелке, $B > 0$ - выбирается малая дуга ($\leq 180^\circ$), $B < 0$ - выбирается большая дуга ($\geq 180^\circ$);

$ALPHA$ - угол наклона полуоси A эллипса к оси x ;

XF, YF - координаты конечной точки дуги.

Если при обращении к программе $ARCELB$ начальная точка совпадает с конечной, то независимо от знаков полуосей вычерчивается полный эллипс с углами наклонов (к полуоси A) начального и конечного радиусов, соответственно, 0° и 360° . Если расстояние между конечной и начальной точками превышает величину диаметра эллипса PQ (рис. 1.11, г), параллельного отрезку, соединяющему эти точки, то обращение к программе $ARCELB$ считается некорректным.

1.6.4. Размерные линии.

Программа $NARROW(XS, YS, XF, YF, S, ICODE)$ чертит отрезок со стрелками на концах.

Параметры программы:

XS, YS - координаты начальной точки отрезка;

XF, YF - координаты конечной точки отрезка;

S - длина стрелки;

$ICODE$ - параметр, управляющий стрелкой: $ICODE = 1$ - стрелка обращена к начальной точке, $ICODE = 2$ - стрелка обращена к конечной точке, $ICODE = 3$ - стрелки указывают на оба конца.

Программа $DIMEN(XS, YS, DATA, SLOPE)$ позволяет изобразить размерную линию заданной длины (рис. 1.12, а). Параметры программы:

XS, YS - координаты начальной точки;

$DATA$ - длина отрезка (размерное число);

SLOPE - угол наклона отрезка к оси x .

Программа DARC($X_C, Y_C, X_S, Y_S, A, ICODE$) позволяет начертить дугу со стрелками на концах. Параметры программы:

X_C, Y_C - координаты центра;

X_S, Y_S - координаты начальной точки дуги;

$|A|$ - угловая величина дуги: $A > 0$ - дуга вычерчивается против часовой стрелки,

$A < 0$ - дуга вычерчивается по часовой стрелке;

$ICODE$ - параметр, управляющий стрелкой: $ICODE = 1$ - стрелка обращена к начальной точке, $ICODE = 2$ - стрелка обращена к конечной точке, $ICODE = 3$ - стрелки указывают на оба конца.

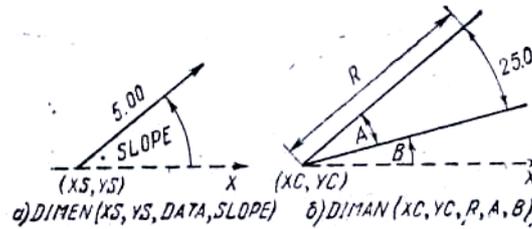


Рис. 1.12. Размерные линии

Программа DIMAN(X_C, Y_C, R, A, B) рисует размерную дугу заданного радиуса. Размерное число надписывается с точностью до одного десятичного знака (рис. 1.12, б). Параметры программы:

X_C, Y_C - координаты центра;

R - радиус дуги;

A - угловая величина дуги;

B - угол между осью x и ближайшей к ней стороной угла.